

JavaScript: orientación a objetos básica



Sergio Luján Mora

Departamento de Lenguajes y Sistemas Informáticos

Universidad de Alicante (España)



Contacto

- <http://gplsi.dlsi.ua.es/~slujan/>
- <http://accesibilidadenlaweb.blogspot.com.es/>
- <http://accesibilidadweb.dlsi.ua.es/>
- <http://desarrolloweb.dlsi.ua.es/>
- <http://www.youtube.com/user/sergiolujanmora>
- sergio.lujan@ua.es
- @sergiolujanmora

¿JavaScript es orientado a objetos?

JavaScript ≠ orientado a objetos

JavaScript = basado en objetos

class



WIKIPEDIA La enciclopedia libre

- Portada
- Portal de la comunidad
- Actualidad
- Cambios recientes
- Páginas nuevas
- Página aleatoria
- Ayuda
- Donaciones
- Notificar un error

- Imprimir/exportar
 - Crear un libro
 - Descargar como PDF
 - Versión para imprimir

- Herramientas
- En otros idiomas

- Afrikaans
- العربية
- Azərbaycanca
- Žemaitėška
- Беларуская
- Беларуская (тарашкевіца)
- Български
- বাংলা
- Български
- Български
- Català
- Česky
- ЧӀаӀауаша
- Dansk
- Deutsch
- Ελληνικά
- English
- Español

Artículo **Discusión**

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,³ basado e dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras dinámicas, en bases de datos locales al navegador...⁴ aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sir se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Contenido [ocultar]

- 1 Denominación e historia
- 2 Referencias
- 3 Bibliografía
- 4 Enlaces externos

Denominación e historia

JavaScript se creó originalmente por Brendan Eich de Netscape con el nombre de *Mocha*, el cuál fue renombrado posteriormente a *LiveScript*. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador 2.003 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.^{5 6}

«JavaScript» es una **marca registrada** de Oracle Corporation. Es usada con licencia por los productos creados por Netscape Communicat Mozilla.⁷

Microsoft dio como nombre a su dialecto de JavaScript «JScript», para evitar problemas relacionadas con la marca. JScript fue adoptado en agosto de 1996, e incluyó compatibilidad con el Efecto 2000 con las funciones de fecha, una diferencia de los que se basaban en esos similares que los términos «JavaScript» y «JScript» a menudo se utilizan indistintamente, pero la especificación de JScript es incompatible con la de ECMA en muchos aspectos.

Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, o Modelo de Objetos del Documento en español), que incorporan Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera la versión 7, Mozilla Application Suite y Mozilla Firefox desde su primera versión.^[*cita requerida*]

JavaScript

Información general

Paradigma	Multi-paradigma, Programación funcional, ¹ Programación basada en prototipos, imperativo, Interpretado (Scripting) .
Apareció en	1995
Diseñado por	Netscape Communications Corp, Mozilla Foundation
Tipo de dato	débil, dinámico, duck
Implementaciones	SpiderMonkey, Rhino, KJS, JavaScriptCore, V8.
Dialectos	ECMAScript
Influido por	Java, Perl, Self, Python, C, Scheme
Ha influido a	ObjectiveJ, JScript, JScript .NET, TIScript

JavaScript	
Información general	
Paradigma	Multi-paradigma, Programación funcional, ¹ Programación basada en prototipos, imperativo, Interpretado (Scripting) .
Apareció en	1995
Diseñado por	Netscape Communications Corp, Mozilla Foundation
Tipo de dato	débil, dinámico, duck
Implementaciones	SpiderMonkey, Rhino, KJS, JavaScriptCore, V8.
Dialectos	ECMAScript
Influido por	Java, Perl, Self, Python, C, Scheme
Ha influido a	ObjectiveJ, JScript, JScript .NET, TIScript

Java	
Desarrollador(es)	
James Gosling & Sun Microsystems <i>Para desarrolladores de Java</i> 	
Información general	
Extensiones comunes	.java, .class, .jar
Paradigma	Orientado a objetos, imperativo
Apareció en	1995
Diseñado por	Sun Microsystems (Oracle Corporation)
Última versión estable	Java Standard Edition 7 (1.7.7) (30 de agosto de 2012; hace 55 días)
Tipo de dato	Fuerte, Estático
Implementaciones	OpenJDK, HotSpot, muchas otras
Dialectos	Generic Java, Pizza
Influido por	Objective-C, C++, Smalltalk, Eiffel
Ha influido a	C#, J#, JavaScript, PHP, Python
Sistema operativo	Multiplataforma
Licencia	GNU GPL / Java Community Process

C++	
Desarrollador(es)	
Bjarne Stroustrup, Bell Labs	
Información general	
Extensiones comunes	.h .hh .hpp .hxx .h++ .cc .cpp .cxx .c++
Paradigma	multiparadigma: orientado a objetos, imperativo, programación genérica.
Apareció en	1983
Diseñado por	Bjarne Stroustrup
Última versión estable	ISO/IEC 14882:2011 (2011)
Última versión en pruebas	C++11
Tipo de dato	fuerte, estático, nominativo
Implementaciones	C++ Builder, clang, Comeau C/C++, GCC, Intel C++ Compiler, Microsoft Visual C++, Sun Studio, Code::Blocks, Zinjai
Dialectos	ISO/IEC C++ 1998, ISO/IEC C++ 2003, ISO/IEC C++ 2011
Influido por	C, Simula, Ada 83, ALGOL 68, CLU, ML ¹
Ha influido a	Perl, LPC, Lua, Pike, Ada 95, Java, PHP, D, C99, C#, Falcon
Sistema operativo	Multiplataforma

JavaScript

Paradigm(s)	Multi-paradigm: scripting, object-oriented (prototype-based), imperative, functional ^[1]
Appeared in	1995; 17 years ago
Designed by	Brendan Eich
Developer	Netscape Communications Corporation, Mozilla Foundation
Stable release	1.8.5 ^[2] (March 22, 2011; 18 months ago)
Typing discipline	dynamic, weak, duck
Major implementations	KJS, Rhino, SpiderMonkey, V8, WebKit, Carakan, Chakra
Influenced by	C, Java, Perl, Python, Scheme, Self
Influenced	ActionScript, CoffeeScript, Dart, JScript .NET, Objective-J, QML, TIScript, TypeScript



La **programación basada en prototipos** es un estilo de **programación orientada a objetos** en el cual, las “clases” no están presentes, y la reutilización de procesos (conocida como herencia en lenguajes basados en clases) se obtiene a través de la clonación de objetos ya existentes, que sirven de prototipos, extendiendo sus funcionalidades. Este modelo es conocido como orientado a prototipos, o programación basada en instancias.

100% orientado a objetos

Ada

Eiffel

Smalltalk

Java

C++

JavaScript



≈ orientado a objetos

Abstracción
Encapsulamiento
Herencia
Modularidad
Principio de ocultación
Polimorfismo
Recolección de basura

Herencia \approx Prototipo

OBJETO

Propiedades
Métodos

Propiedades ESTADO

COMPORTAMIENTO

Métodos

**Casi Todo en JavaScript
es un objeto**

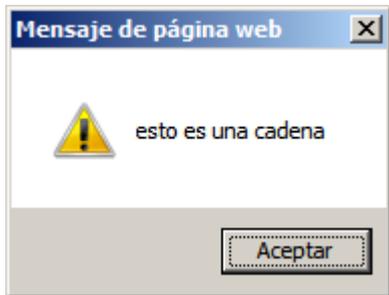
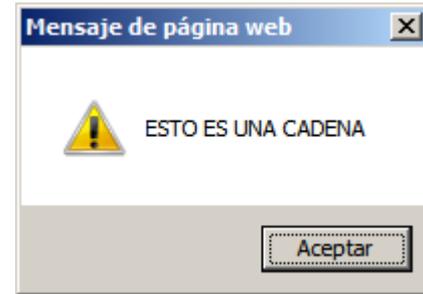
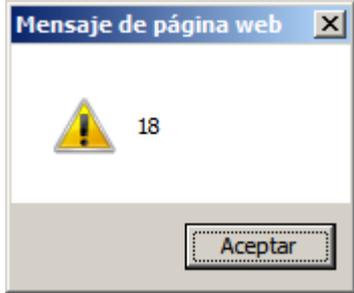
Esto es un objeto
de tipo String

```
var a = "Esto es una cadena";
```

```
alert(a.length);
```

```
alert(a.toUpperCase());
```

```
alert(a.toLowerCase());
```





```
a["length"]
```

PROPIEDAD a.length

MÉTODO a.toUpperCase()



Array

Date

Math

Number

String

RegExp



<http://idesweb.es/>

idw@idesweb.es

@idesweb